Status Report: Quality-Based Reliable Computing

W. Kent Fuchs, Ward Page, William H. Sanders
School of Electrical and Computer Engineering - Purdue University
Naval Command, Control, & Ocean Surveillance Center, RDT&E
CRHC - Coordinated Science Laboratory - University of Illinois
http://www.crhc.uiuc.edu/PERFORM/QBRC.html

W. Kent Fuchs
Nuno Neves
Kuo-Feng Ssu
Chen Wang
Bin Yao

William Sanders
Dan Deavours
Jay Doyle
G. P. Kavanaugh
Doug Obal
John Sowder
Alex Williamson

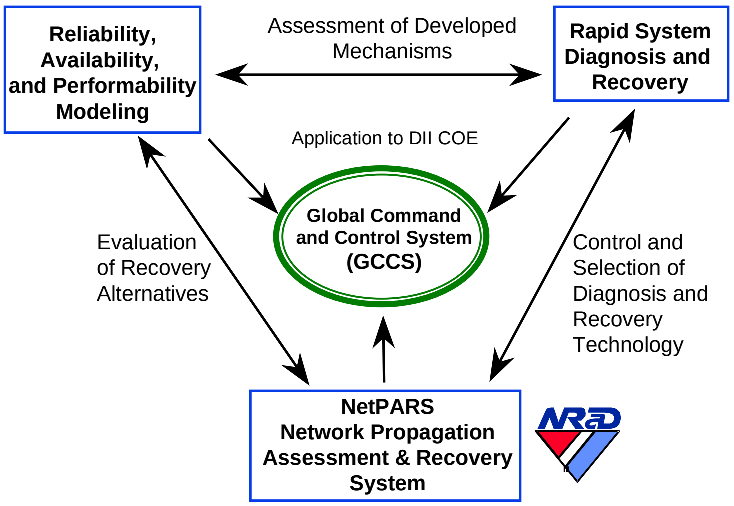
Ward Page
Peter Kaomea
Chris Acantilado
Mike Kalman

Project funded by DARPA ITO under Fort Huachuca Contract DABT63-96-C-0069

Quality-Based Reliable Computing

W. Kent Fuchs, William H. Sanders, Ward Page

Approach:



New Ideas:

- **New mechanisms for achieving high availability and dependability** in operational command and control computing
- **New modeling and evaluation technologies** to assess quality of fault-tolerant mechanisms in operational command and control environment
- **New methodologies for evaluating the effects of faults on the quality of information** used in command and control decision making

Schedule:

Sept. '96

- Developed techniques for real-time preemptive diagnosis.
- Developed quality propagation language (QPL) specification.
- Developed model specification language and state-based solver
- Developed first-generation NetPARS system and agent architecture.

Sept. '97

- Develop network-based recovery, detection and diagnosis software.
- Extend rapid recovery and early diagnosis techniques to incorporate heterogeneous processor, server, and network nodes.
- Develop fault simulator, integrate QPL with simulator and solver
- Integrate QPL and truth maintenance system into NetPARS.
- Develop fault-detection and base quality collection agents.

Sept. '98

- Develop recovery techniques for high-performance mobile computers that incur sustained stress and massive failures.
- Integrate modeling tool with NetPARS, apply to command and control application
- Develop backward propagation and recovery tools for NetPARS.
- Deploy results in a distributed command and control testbed.

Sept '99

Impact:

New System Diagnosis and Recovery Techniques:
- Increase command and control system reliability and availability

New Modeling and Assessment Tools:
- Accurately account for complex dependencies present in command and control applications
- Solve models in space- and time-efficient manner

Network Propagation, Assessment, and Recovery System:
- Identifies information qualities affected by fault propagation
- Traces the path of complex faults to their origin

Purdue University - University of Illinois - Naval Command, Control, and Ocean Surveillance Center

Objective

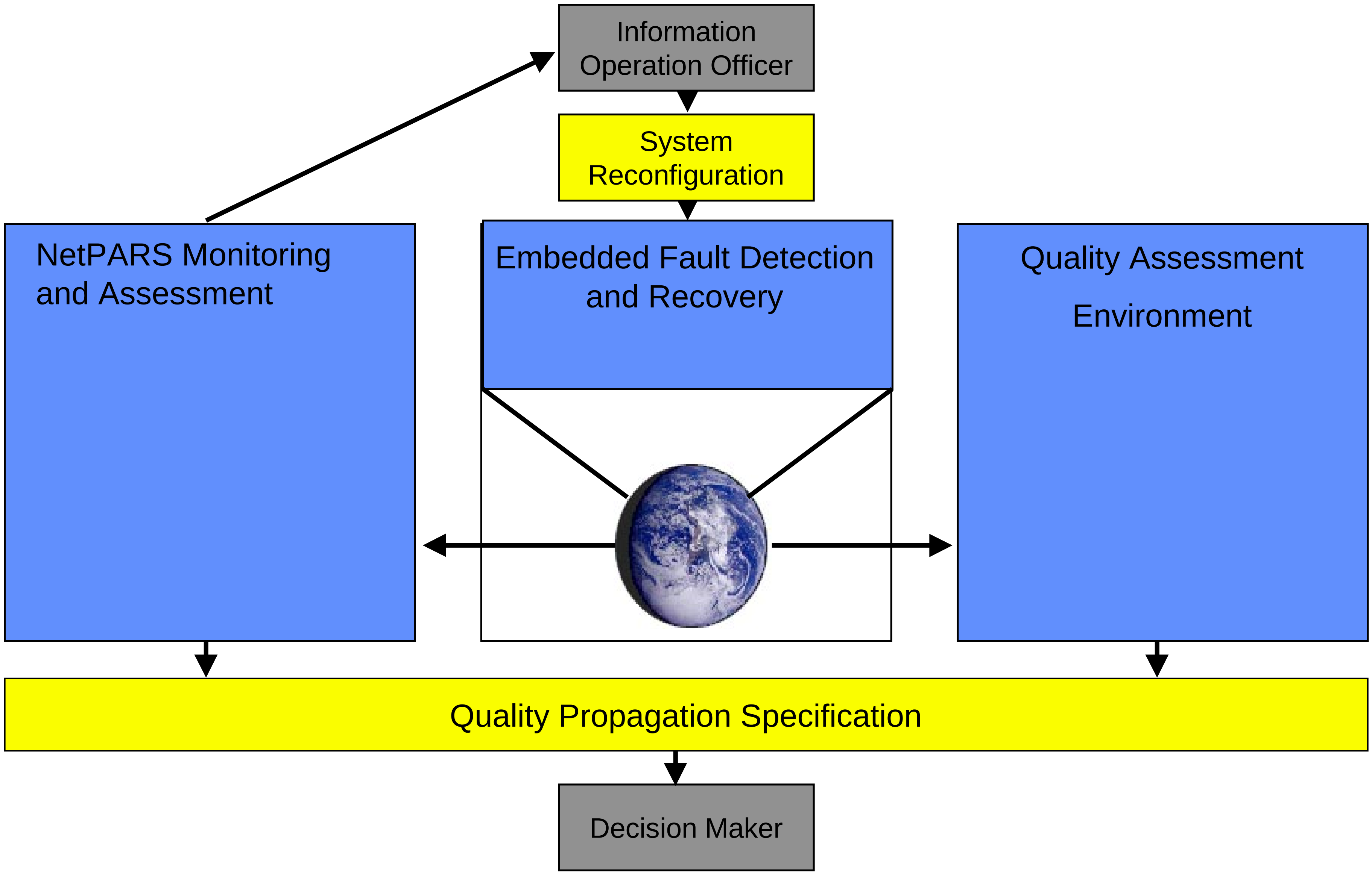
⇒ **Highly Reliable Information Technology for the Military that Is Failure, Application, and System Comprehensive.**

- Failure Comprehensive:
  Both Intentional and Accidental Faults
- Application Comprehensive:
  Use characteristics of application for rapid recovery
- System Comprehensive:
  Is multilevel, with different recovery strategies at each level.

Key Idea: Use the quality of information, as presented to a decision maker, as the measure of dependability.

On and off-line modeling and analysis will be developed to insure that the desired qualities are achieved.

QBRC System Overview



Agenda

- 11:20 - 11:25 **Overview of project** -- Sanders
- 11:25 - 11:35 **Next-Generation Dependability Prediction Tool** -- Sanders
- 11:35 - 11:45 **Network Propagation, Assessment and Recovery** -- Page
- 11:45 - 11:55 **Failure Recovery in Networks of Workstations** - Fuchs
- 11:55 - 12:00 **Questions**

Next-Generation Dependability Prediction Tool

William H. Sanders
CRHC - Coordinated Science Laboratory
University of Illinois at Urbana Champaign
whs@crhc.uiuc.edu
http://www.crhc.uiuc.edu/PERFORM/QBRC.html

W. Kent Fuchs
Nuno Neves
Kuo-Feng Ssu
Chen Wang
Bin Yao

William Sanders
Dan Deavours
Jay Doyle
G. P. Kavanaugh
Doug Obal
John Sowder
Alex Williamson

Ward Page
Peter Kaomea
Chris Acantilado
Mike Kalman

Project funded by DARPA ITO under Fort Huachuca Contract DABT63-96-C-0069

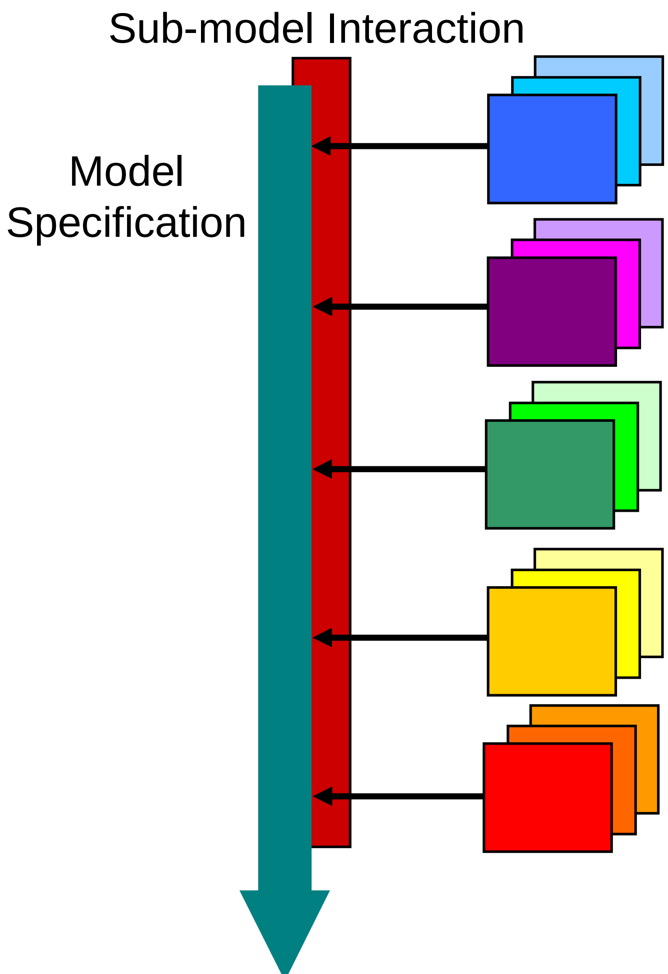
Goal

- **System-comprehensive modeling and dependability assessment of fault-tolerant mechanisms, networks, hardware, operating systems, middleware, and applications.**
- Challenges:
  - Complexity, both in representing complex behaviors, and in solving the resulting models
  - Composibility, to build larger models out of submodels, where the submodels interact with each other in various ways, including changing each others, state, or passing results.
  - Extensibility, to permit easy addition of new model specification and solution methods

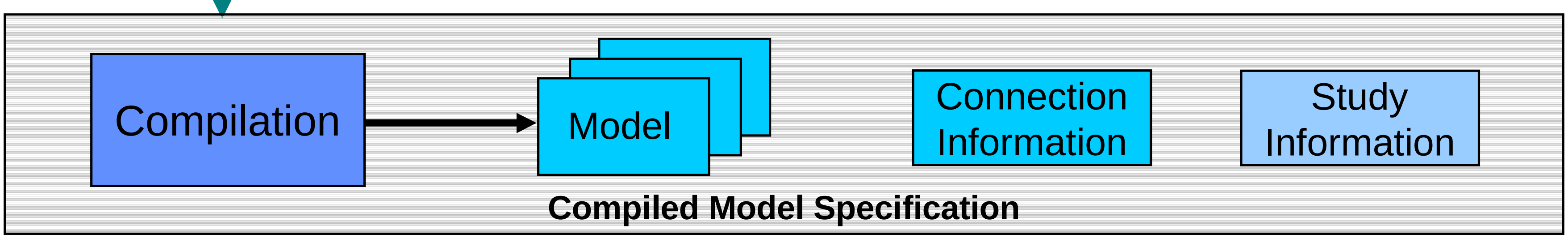
Approach

- Multiple Atomic Specification Languages
  - Different parts of the system will be specified using different model specification languages. This will permit parts of a model to be specified in a formalism that is natural for the subsystem being considered.
- Composition of multiple, heterogeneous models through simple, functional interfaces
- Connection of multiple interacting models that are solved independently, and interact by passing results (e.g., single number, distribution, or abstract state model)
- Multiple simulation and analytic based model solution engines that are written in a formalism independent way
- Abstract functional interaction between model/model and model/solver permits rapid integration of new formalisms and model solution engines into the tool

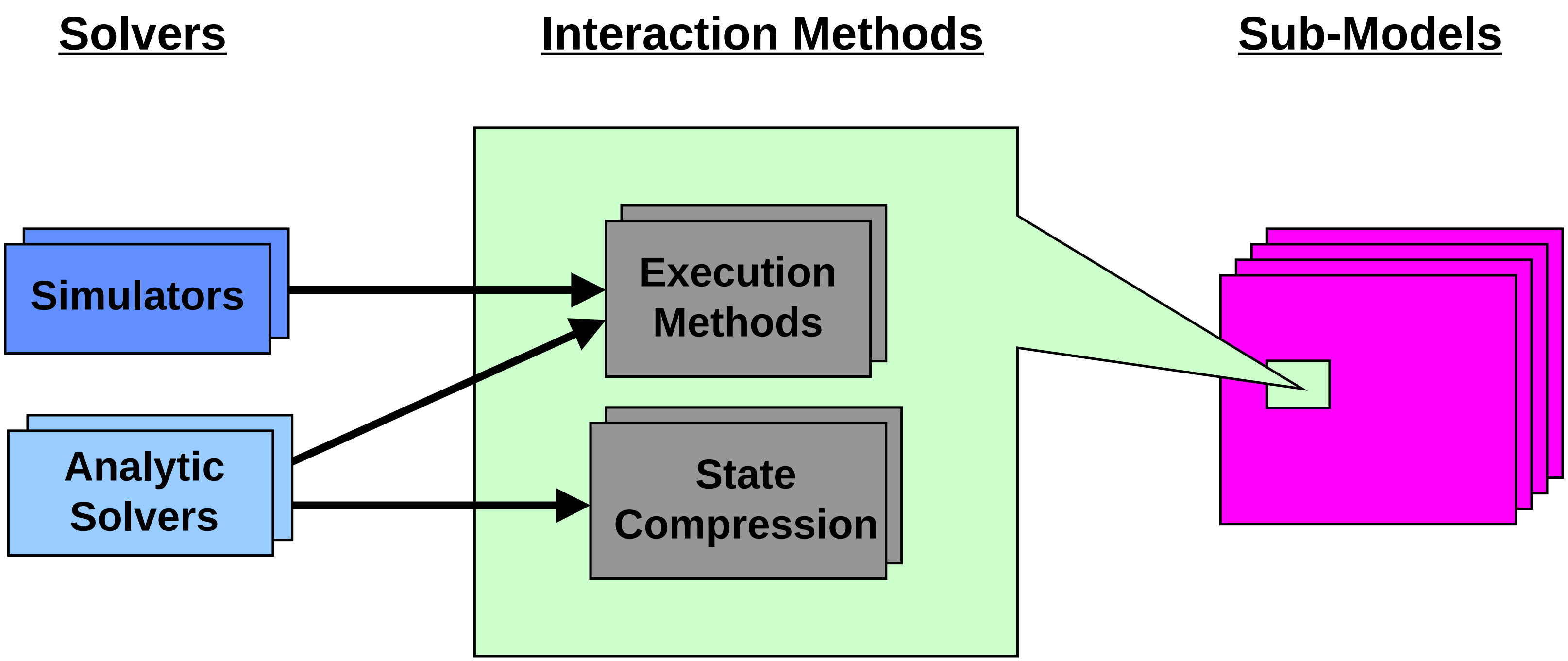
Overview of Model Construction



Constructor	Example Specification Language
Model Specification	Stochastic activity network High-level Petri net Queuing network
Model Composer	Rep-Join tree Graph interconnection
Variable Specifier	Quality propagation language Rate and impulse reward variable
Model Connector	Fixed-point governor Acyclic model composer
Study Editor	Range and Set Variation Non-linear optimizer

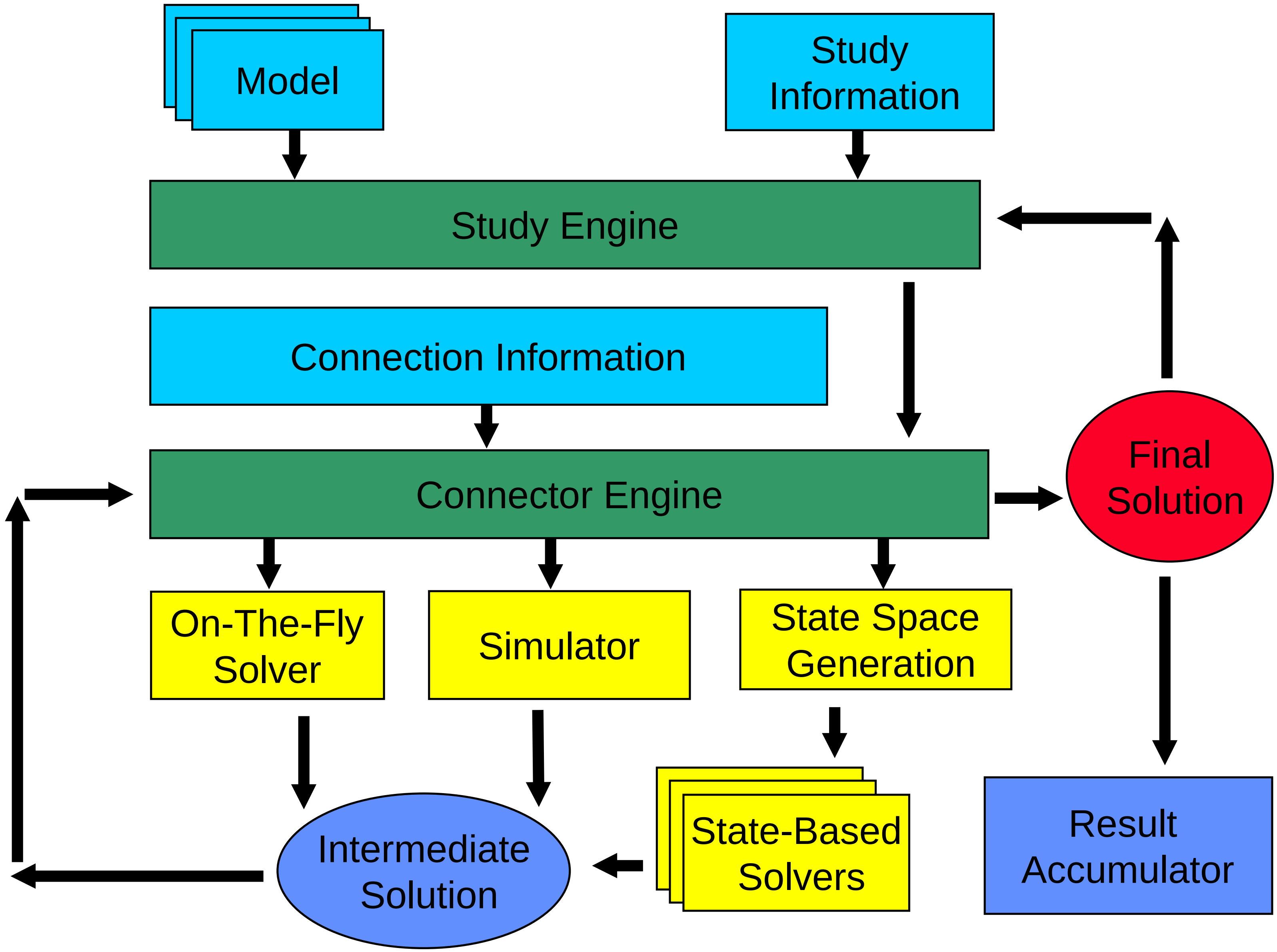


Abstract Method-Based Interface Simplifies Interaction Between Models and Solution Engines



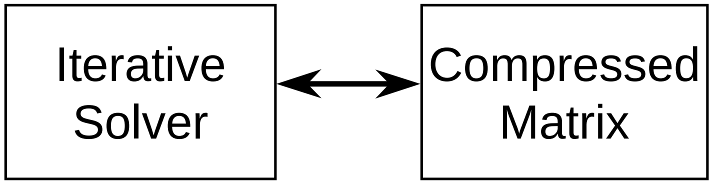
- Small set of formalism-independent method declarations permit submodels to interact with solvers in abstract way
- Formalism-independent method declarations permit solvers to act on compact state representations without understanding details
- Formalism-specific implementations provide appropriate time/ space tradeoffs and efficient execution of solvers.

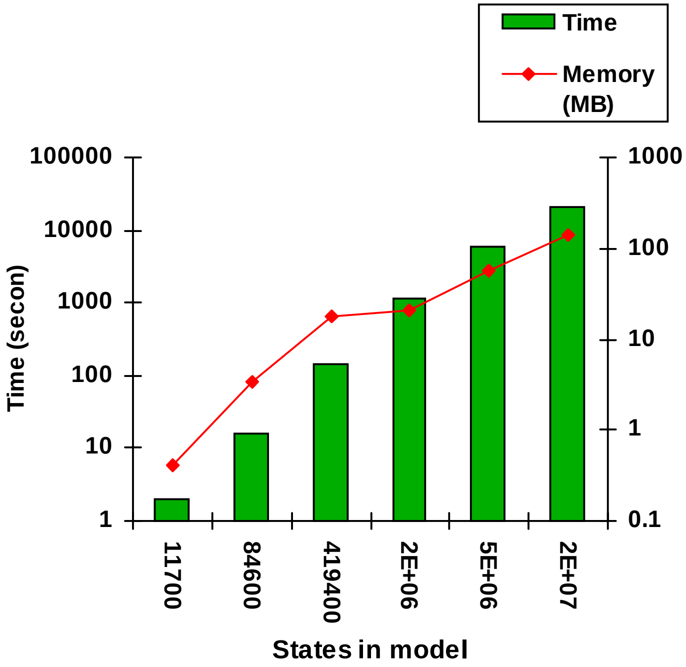
Model Solution



Example Solver: Largeness-Tolerant Steady-State Solution Techniques

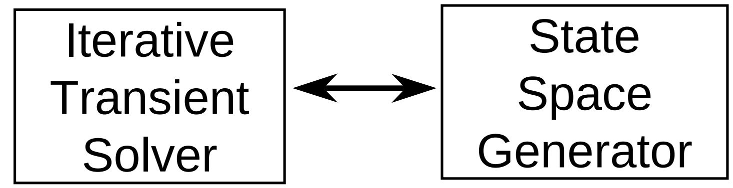
- Numerical solutions to millions of states
- On-the-fly methods
  - Model is compressed matrix
  - Gauss-Seidel with forward *or* backward search
  - Solution techniques with locality
- Disk-based methods
  - Block Gauss-Seidel
  - High data generation rate (5 MB/s) with low CPU overhead (< 20%)
  - Memory: solution vector + buffers
  - Most promising so far





Example Solver: On-the-Fly Transient Solution Techniques

- Analytical solution to models with a large number of states
- Combines state space generation and iterative transient techniques to reduce state space
- Reduces number of multiplications by operating only on active states
- Efficient memory usage by exploiting sparse matrix storage techniques
- Example: SNARC fault tolerant multiprocessor (Lee, Abraham, Rennels, and Gilley, DCCA 2)



Time Instant 1 Year

	Number of Processors			
Error Bound	1	2	3	4
10^{-3}	56	724	11870	57882
10^{-6}	56	724	28103	103446
10^{-9}	68	1432	56540	331589
10^{-12}	68	2366	64789	711826
Full State Space	74	2775	70300	$10^{\wedge 6}$ +

Number of States

Recent Accomplishments

- Defined quality propagation language, to translate base qualities, as reflected by system status, to the quality of data presented to a decision maker.
- Developed abstract, functional interfaces to facilitate interaction between submodels, composers, and performance variable specifiers
- Constructed prototype model and quality interface builder
- Constructed on-the-fly state explorer and transient solver
- Constructed disk-based steady state solver, that operates efficiently on a compact disk representation of a Markov model
- Constructed on-the-fly steady state solver, that operates directly on model representation, rather than a state-level model
- Completed the initial design, and are beginning the development of a formalism-independent discrete-event simulator

Dependability Assessment Tool Release Status

- Initial version currently under development -- release scheduled for January, 1998.
  - Single model specification method - stochastic activity networks
  - Single composer
  - Reward variable specifier
  - Simulator, on-the-fly solver, disk-based steady-state solver
- Second release April 1998.
  - Second model specification method
  - Second, graph-based, composer
  - Quality propagation specifier
  - Multi-machine simulator

Network Propagation, Assessment, and Recovery

Ward Page
Naval Command, Control, & Ocean Surveillance Center, RDT&E
Advanced Decision Technology Group
San Diego, CA
page@nosc.mil
http://www.crhc.uiuc.edu/PERFORM/QBRC.html

W. Kent Fuchs
Nuno Neves
Kuo-Feng Ssu
Chen Wang
Bin Yao

William Sanders
Dan Deavours
Jay Doyle
G. P. Kavanaugh
Doug Obal
John Sowder
Alex Williamson

Ward Page
Peter Kaomea
Chris Acantilado
Mike Kalman

Project funded by DARPA ITO under Fort Huachuca Contract DABT63-96-C-0069

Technical Objectives

- Map & report in real-time the propagation of degraded data quality in information networks
  - concentrate on faults in the data layer
  - faults can occur as a result of:
    - component (hardware) failures
    - poor system design
    - poor implementation
    - bad system usage - non-malicious
    - bad system usage - malicious
- Locate fault source(s)
- Provide tools to select appropriate recovery mechanism(s)
  - optimize resource allocation

Technical Approach

- Model the data layer and the processes that affect the data
- Model the data qualities that can be affected
- Map the real world info system & monitor for faults
- Forward propagate faults through the data layer & report effects on data
- Backward propagate faults from data layer through software and hardware layers to create possible fault scenarios & build list of possible fault sources (with likelihood estimates)
- Suggest recovery method(s) for fault recovery

System Users

Primary concern: Information System Diagnostics (ISDs)

System Administrator (e.g., CSOW)

- **System supervisors**
- **System operators**
- **System technicians**

Primary concern: Data Quality Indicators (DQIs)

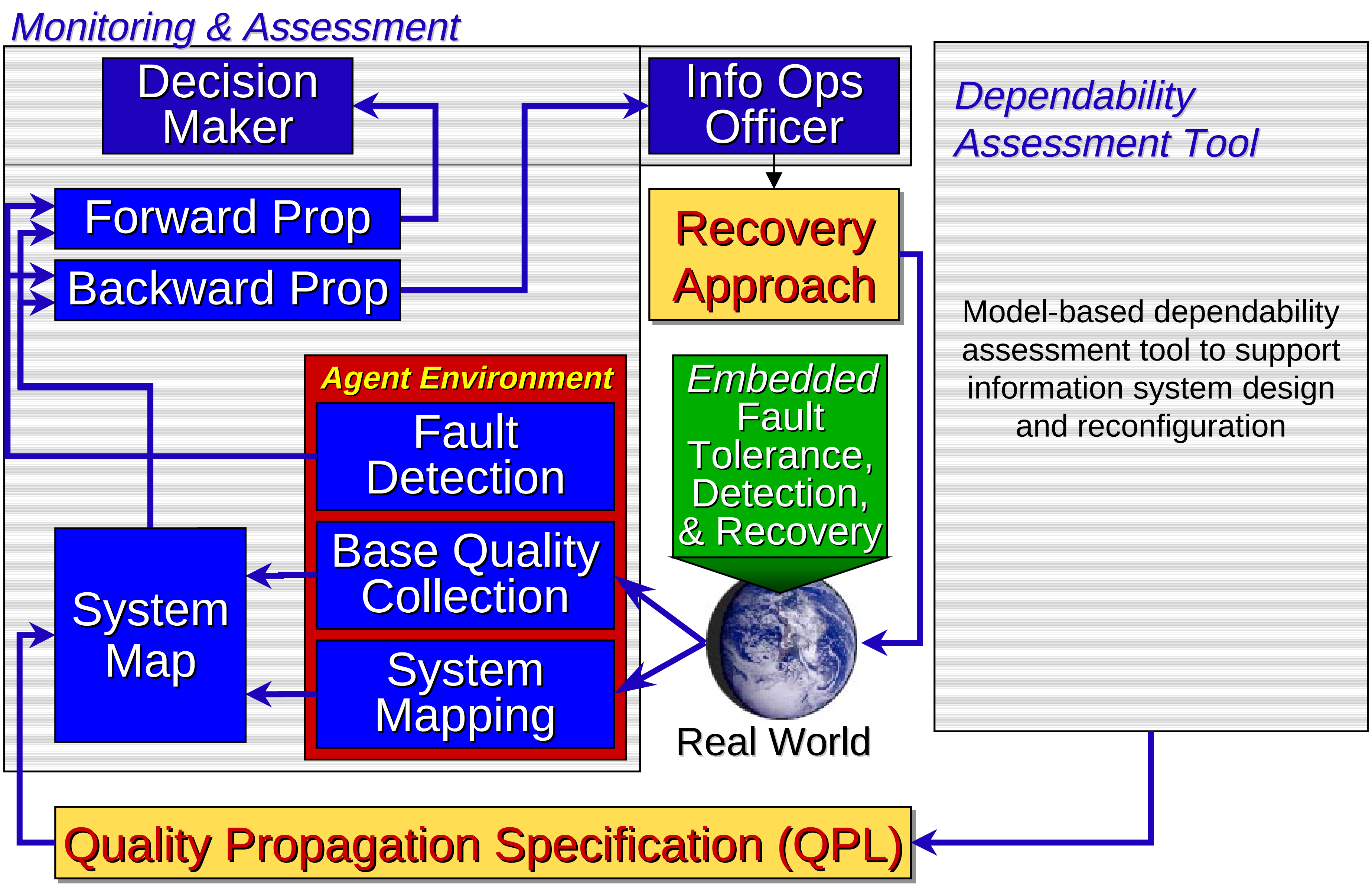
End-user / Decision Maker (e.g., CO & TAO)

- **Watch Officer**
- **Watch standers**
- **Sensor operators**

System Decomposition

- System design and prediction
  - Stochastic models of failure rates & modes
- System monitoring
  - Embedded fault tolerance mechanisms
  - Monitoring and detection agents
- Fault assessment (effects and sources)
  - Forward and backward rule-based propagation
- Recovery
  - Embedded mechanisms
  - Applied mechanisms
    - Roll back recovery
    - Application software
    - Hardware replacement

System Architecture



Data Quality Taxonomy

	Data Transforms		Data Qualities	
<u>Fourth level:</u> **Recovery Processes**	• re-route • annotate (DQIs) • repair • re-ID/re-interpret		• credible • believable • commonly understood	
<u>Third level:</u> **Logical processes**	• route • recall • excerpt • ID/interpret		• relevance • communality • commonality	• confidentiality • (un)ambiguity • meaningfulness
<u>Second level:</u> **Multiple data units**	***Input-Output*** • aggregate • reduce • disaggregate	***Output-Output*** • replace • update	• completeness • currency	• full correctness - error rate - precision
<u>First level:</u> **Single data units**	*Interfaces* • observe • present • delete • destroy	*Communication* • transport • function - copy - other	• availability / timeliness • correctness	

Mapping & Fault Detection Agents

- Manual -- Major data flows in systems can be manually entered into NetPARS. These types of flows are generally available at design.
- Automatic -- Actual system assets (hardware, software, data) can be detected & network connectivity and routing can be mapped.
- Availability -- Agents are placed at key points of vulnerability to detect if data is at a specified location at a specified time.
- Correctness -- Agents compare available data to redundant data sources and/or integrity constraints to assess correctness.
- Relevance -- Users are asked to enter what data is relevant to their work. (Agents may later be used to determine a baseline for relevance by watching what data users ask for.)

Failure Recovery in Clusters of Workstations

W. Kent Fuchs

School of Electrical and Computer Engineering

Purdue University

fuchs@purdue.edu

http://www.crhc.uiuc.edu/PERFORM/QBRC.html

W. Kent Fuchs
Nuno Neves
Kuo-Feng Ssu
Chen Wang
Bin Yao

William Sanders
Dan Deavours
Jay Doyle
G. P. Kavanaugh
Doug Obal
John Sowder
Alex Williamson

Ward Page
Peter Kaomea
Chris Acantilado
Mike Kalman

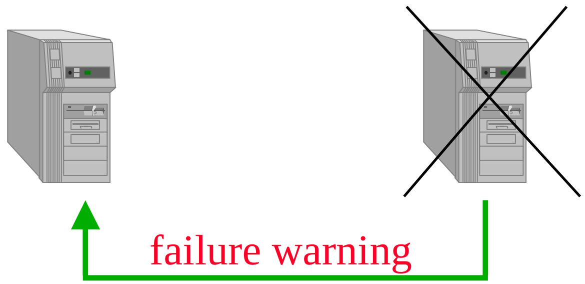
Project funded by DARPA ITO under Fort Huachuca Contract DABT63-96-C-0069

Fault Detection and Diagnosis

- Use the error codes returned by the stream sockets to locate process failures.
- No failure-free overheads since the errors are generated automatically.
- Types of failures:
  - Kill : terminates the process but leaves the rest of the system undisturbed.
  - Reboot : machine shuts down, and then boots.
  - Crash & boot : machine crashes and then boots.
  - Crash : machine crashes permanently or stays down for a long period of time.

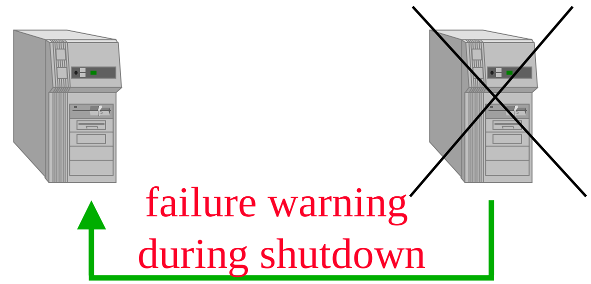
Fault Detection Using Hints From Sockets

Kill



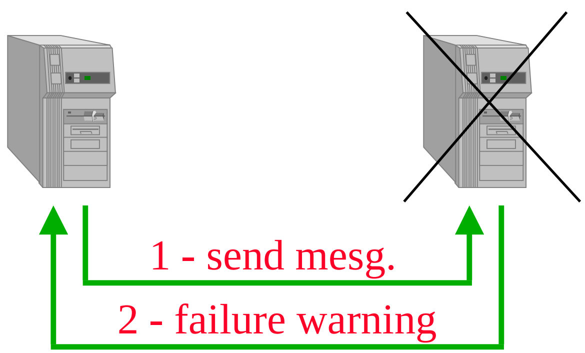
Latency : from a few milliseconds

Reboot



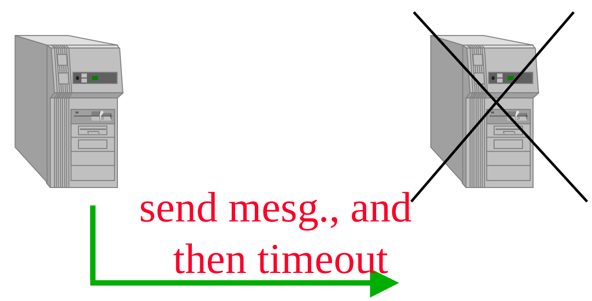
Latency : around 3.7 seconds

Crash & boot



Latency : around 97 seconds

Crash



Latency : around 490 seconds

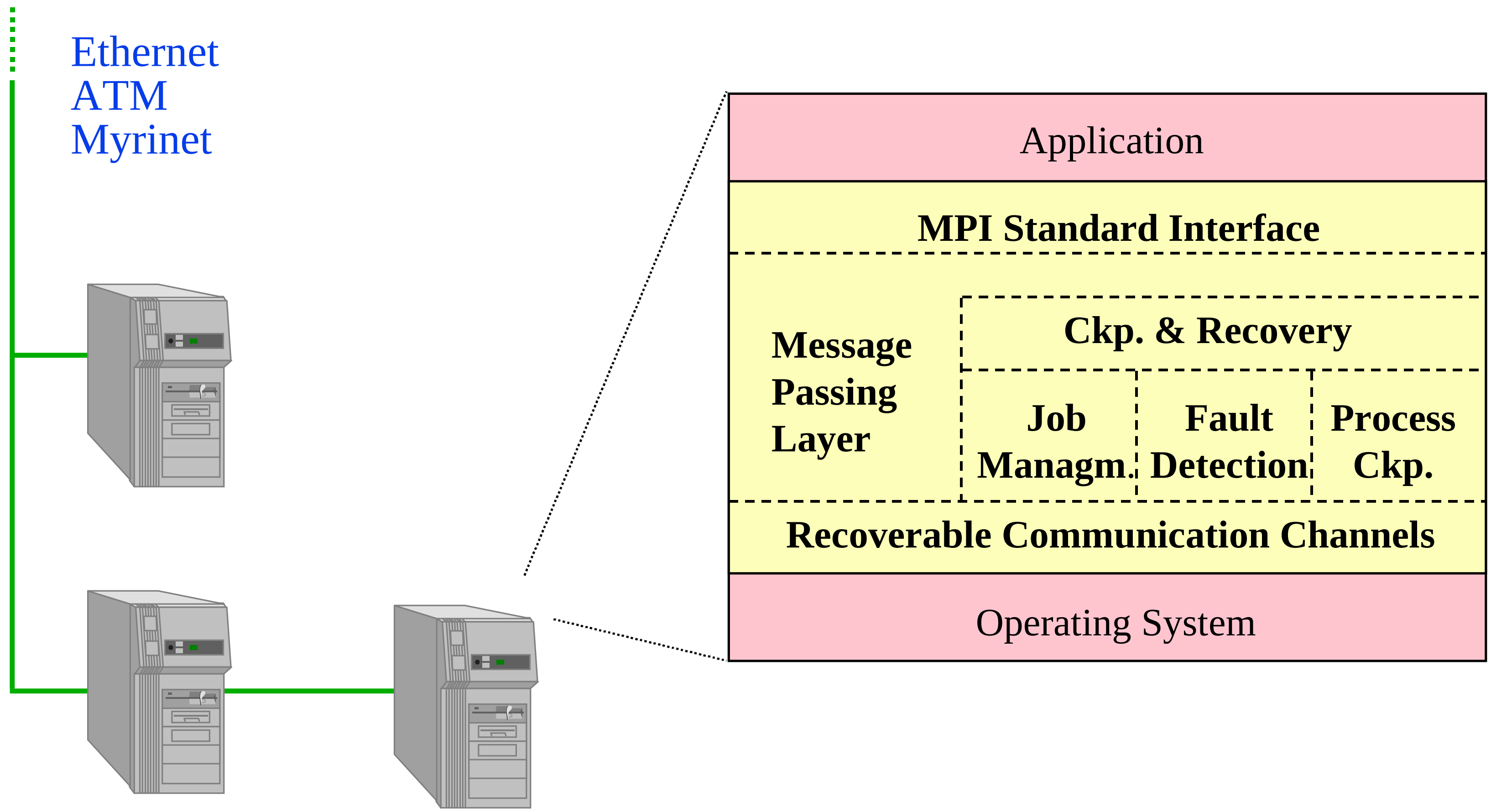
The Two Sides of RENEW

RENEW
REcoverable NEtwork of Workstations

Message passing system
for clusters of workstations

Tool that allows rapid
implementation of checkpoint
protocols

RENEW and its Environment



Experimental Results

- New time-based checkpoint protocol with logging at sender: Uses time to indirectly coordinate the storage of the checkpoint
  - No message sends during checkpoint storage
  - No need for message tagging
  - No extra accesses to disk beside checkpoint storage
- Applications:
  - ga : parallel genetic algorithm
  - ising : particle simulator
  - povray : parallel raytracer

8 HP workstations over 10 Mbits/s Ethernet

App	No Ckp	Time-Based Ckp				Overh
	sec	NCkp	size [KB]	Ckp Time	sec	%
ga	2851	9	811/585	4.3/3.6	2901	**1.7**
ising	3147	10	2490	16.8	3281	**4.2**
povray	2542	8	694/2157	5.7/13.4	2645	**4.0**

Ckp Period is 5 minutes; Ckp written to HP Workstation

4 Sun Workstations with 155 Mbits/s ATM

App	No Ckp	Time-Based Ckp				Overh
	sec	NCkp	size [KB]	Ckp Time	sec	%
ga	1514	5	839/648	1.6/1.3	1517	**0.2**
ising	2841	9	4536	9.3	2954	**4.0**
povray	1804	6	595/2165	1.2/4.3	1855	**2.8**

Ckp Period is 5 minutes; Ckp written to local disk

App	No Ckp	Time-Based Ckp				Overh
	sec	NCkp	size [KB]	Ckp Time	sec	%
ga	1514	4	839/649	0.3/0.4	1454	**---**
ising	2841	9	4537	2.8	2874	**1.2**
povray	1804	6	595/2045	0.4/1.0	1829	**1.4**

Ckp Period is 5 minutes; Ckp written to remote HP Workstation

Future Work: Failure Recovery in Clusters of Workstations

- Integrate the fault detection mechanism with the routing protocols of high-throughput networks to improve the detection latency of crash faults.
- Complete the definition of the standard interface for constructing checkpoint protocols.
- Implement several checkpoint protocols and create a database of recovery techniques.
- Expand the environment of RENEW to include mobile hosts.